
TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: B2612 – Elektrotechnika a informatika

Studijní obor: 2612R011 – Elektronické informační a řídicí systémy

Bakalářská práce

Realizace telefonního dialogového systému
Realization of phone dialogue system

Autor:

Vedoucí bakalářské práce:

Konzultant bakalářské práce:

Zdeněk Mauer

Ing. Miroslav Holada, Ph.D.

Ing. Josef Chaloupka, Ph.D.

V Liberci dne 3.5.2009

Sem vložit zadání

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom(a) toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum:

Podpis:

Abstrakt

Tato bakalářská práce se zabývá problematikou telefonních dialogových systémů s využitím hlasového rozpoznávače a to nejen po teoretické, ale i praktické stránce při konečné realizaci a odzkoušení systému. Za tímto účelem byl vytvořen program v jazyce C++ sloužící k ovládání telefonní karty Dialogic D/4PCI. Program umožňuje ovládání pomocí jednoduchého rozhraní, které bylo použito i při testování.

Pro interakci systému s volajícím je nutné využít hlasový rozpoznávač, se kterým program komunikuje a rozpoznávač zasílá zpět slova v textové podobě. Byla použita distribuovaná varianta rozpoznávače komunikující pomocí TCP/IP protokolu. Zkušební verze systému poskytuje informace o některých telefonních číslech zaměstnanců ústavu ITE. Rozšíření o více kontaktů, nebo zakomponování určitého způsobu dalšího větvení dialogu je možné.

V první části práce je krátká charakteristika použitého hardwaru, shrnutí principu některých standardů, jako například zvukových formátů a také shrnutí vlastností knihoven telefonní karty D4/PCI. Dále jsou popisovány principy rozpoznávání hlasu a hlasové syntézy.

Další část je věnovaná přímo samotné realizaci. Obsahuje opis některých částí zdrojového kódu s komentáři k vysvětlení principu činnosti systému, textovou ukázkou komunikace a celkový vývojový diagram programu.

V závěru této práce je shrnutí výhod a nevýhod zvoleného řešení, nástin jiných možností a zdůvodnění konečné podoby.

Klíčová slova: Telefonní dialogový systém, distribuovaný hlasový rozpoznávač, syntéza řeči, Dialogic D/4PCI, A-law, μ -law, PCM, TCP/IP

Abstract

This bachelor work deals with questions of phone dialogue systems with using voice recognizer respectively not only on theoretical side, but also on practical side on final realization and testing of system. For that purpose was created program in C++ language performed for control of phone card Dialogic D4/PCI. Program makes possible control per simple interface, which has been use on testing.

For interaction of system on caller is needed use voice recognizer, with that program communicates, and recognizer send back words in text form. It was used distributed version of recognizer, communicating per TCP/IP protocol. The demo application of system provides information about some telephone numbers of members on ITE institute. Extension of next contacts or composition of specific method of dialogue branching are possible.

First part of work contains short specification of used hardware, some standards principle summary, as sound formats as well as summary properties of phone card D4/PCI libraries. Description of voice recognizing and voice synthesizes principles are below.

Next part is focused on realization of application. It contains description of parts of source code with comments that explains principle of system work, text example of communication and global schema and documentation of program.

The conclusion of this work summarizes advantages and disadvantages of chosen solution, proposal of other possibilities and motivation of final form.

Keywords: Phone dialogue system, voice recognizer, voice synthesis, Dialogic D4/PCI, A-law, μ -law, PCM, TCP/IP

Obsah

Abstrakt.....	4
Abstract.....	5
Obsah	6
Obsah	6
Úvod.....	7
1. Telefonní karta Dialogic D/4PCI.....	9
1.1. Knihovna Voice API.....	9
1.2. Popis funkcí Voice Library	10
1.3. Asynchronní a synchronní režim	12
2. Formáty pro záznam zvuku.....	14
2.1. Pulzně kódová modulace PCM.....	14
2.2. Logaritmická komprese A-law, μ -law	15
2.3. Kódování DPCM, ADPCM	16
2.4. Použití v dialogovém systému	16
3. Distribuovaný hlasový rozpoznávač	19
3.1. Základní princip	19
3.2. Použitý rozpoznávač izolovaných slov	20
3.3. Implementace rozpoznávače	21
4. Syntéza řeči.....	22
4.1. Lineární modely	22
4.1.1. Formantový model	23
4.1.2. LPC model	23
4.1.3. Kepstrální model.....	23
4.2. Sinusový model.....	23
4.3. Syntéza skládáním řečových úseků, PSOLA.....	24
4.4. Použitý modul syntézy - Modul TTS.....	24
5. Návrh telefonního dialogového systému	25
5.1. Typ dialogového systému	25
5.2. Diagram telefonního dialogového systému	25
5.3. Test na zavěšení	27
5.4. Slovník.....	28
6. Testování systému.....	29
6.1. Připojení ke vzdálené ploše	29
6.2. SIP telefonie v síti CESNET2.....	30
6.2.1. Síť CESNET2	30
6.2.2. VoIP telefonie	30
6.2.3. SIP protokol	31
6.3. X-lite 3.0	33
6.4. Nahrazení telefonní karty SIP klientem	36
6.4.1. Pasivní připojení SIP klienta.....	36
6.4.2. Aktivní připojení SIP klienta	36
6.5. Přepojování	36
7. Závěr	38
8. Zdroje.....	39
Příloha A	40

Úvod

V dnešní době se snad každý setkal s nějakým telefonním dialogovým systémem. Běžně tyto systémy využívají různé instituce v bankovním sektoru pro sdělení základních informací. Díky těmto systémům lze určit oblast, na kterou se chce volající dotazovat a následně ho přepojit k pracovníkovi, který je schopen nejkvalitněji dané dotazy zodpovědět. Většina těchto systémů pracuje s DTMF tóny. Volající tak pomocí klávesnice svého telefonu prochází strukturou daného systému k informacím, které ho zajímají.

Zadáním této práce je realizace dialogového telefonního systému s využitím distribuovaného rozpoznávače. Podobný systém (Infocity) již byl vytvořen na Technické univerzitě v Liberci a fungoval již několik let. Práce byla zadána kvůli rozvoji novějších technologií, které v době realizace projektu Infocity nebyly k dispozici.

Původní systém ovládal telefonní kartu Dialogic shodně pomocí C++, samotný systém byl vytvořen v prostředí VisualBasic a větvení dialogu bylo realizováno v prostředí Lotos. Zejména posledním jmenovaným byl systém velmi nekompaktní a omezený v možnostech rozšíření, protože Lotos je uzavřený systém. Zejména proto bylo rozhodnuto o určitém kroku zpět, ve kterém dojde k přepracování celého systému v prostředí C++. Další rozšiřování je možné. Díky rozvoji nových technologií lze použít místo distribuovaného rozpoznávače izolovaných slov třeba rozpoznávač spojitě řeči. Zároveň je možné používat dokonalejší distribuované systémy pro syntézu řeči.

Základem pro konstrukci systému je ovládání telefonní karty Dialogic v prostředí C++, což bylo časově hlavní náplní práce. Ovládání je možné v synchronním, nebo v asynchronním režimu. Po konzultaci s vedoucím bakalářské práce bylo rozhodnuto pro realizaci v asynchronním režimu. Při této variantě bohužel nefungovaly některé funkce tak, jak je přímo popsáno v manuálu k zařízení, takže bylo nutné uchýlit se k některým řešením, které nejsou příliš elegantní, ale jsou funkční. Zejména tato skutečnost byla důvodem největšího zdržení při realizaci systému.

Velkou inovací byl způsob, jakým byla práce tvořena. K testování byl samozřejmě potřeba specifický hardware (karta Dialogic), který není standardně ve

výbavě běžného osobního počítače. Aby bylo možné pracovat na systému kdekoliv, byl vytvořen účet na osobním počítači ústavu ITE, označovaným jako DOK02, který je tímto hardwarem vybaven. Pomocí aplikace „Připojení ke vzdálené ploše“, která je standardní součástí systému MD Windows XP, probíhalo ovládání počítače připojeného k síti v budově B Technické univerzity v Liberci, ke kterému nebyl nutný fyzický přístup.

Další inovace byla v systému testování telefonního dialogu. Telefonní karta Dialogic byla připojena k telefonní ústředně univerzity a měla přiřazené telefonní číslo 3100. Volání v rámci této ústředny bylo samozřejmě možné z jakéhokoli aparátu zdarma, ale v případě testování z místa mimo TUL by bylo nutné pomocí volat předvolby na číslo +420485353100 a hovor platit. Z tohoto důvodu byl zřízen účet u společnosti CESNET, s jehož pomocí bylo možné volat pomocí klienta X-lite 3.0 do sítě TUL zdarma.

Zde se zrodila i myšlenka dalšího rozšíření dialogového systému, kdy by místo telefonní karty bylo použito IP telefonie. Dialogový systém by poté bylo možné využívat stejně jako nyní. Proto je tato práce věnována i základní problematice IP telefonie, SIP protokolu, síti CESNET2 a další poskytovatelům v služeb v této oblasti.

1. Telefonní karta Dialogic D/4PCI

1.1. Knihovna Voice API

Tato knihovna tvoří tělo této bakalářské práce. Její pomocí je zajišťován chod veškerých hlasových procesů. Jako hlavní pomoc při práci s touto knihovnou sloužil soubor nápovědy `voxapi95.hlp`, která je zároveň hlavním zdrojem pro tuto práci. Začátek je věnován základnímu označení telefonních karet Dialogic. Byly vyráběny ve 2 a 4 kanálové variantě, odtud tedy označení D/4PCI. Pro aplikaci byl použit port č.4, protože některé z portů jsou poškozeny z dřívějších používání. Použití vytvořené aplikace pro kartu D2/PCI, nebo pro jiný port karty D/4PCI tedy není možné bez změny ve zdrojovém kódu.

Průvodce je rozdělen do několika částí.

- Kapitola 1 poskytuje náhled na hlasový software. Jedná se o seznamy komponent a podporovaných karet Dialogic a popisuje hlasové ovladače a hlasové knihovny.
- Kapitola 2 poskytuje základní informace o hlasové knihovně `libdxxxmt.lib` a `libdxxxmt.dll`. Poskytuje náhled na funkční kategorie, a popisuje požadavky programování k použití knihovny.
- Kapitola 3 poskytuje kompletní seznam funkcí (v abecedním pořadí) pro všechny funkce ve zvukové knihovně.
- Kapitola 4 popisuje následující témata:
 - o Datové struktury a tabulky obsažené v hlasové knihovně.
 - o Parametry definované pro hlasová zařízení která mohou být nastavována nebo Získány použitím `dx_getparm()` a `dx_setparm()`.
- Kapitola 5 obsahuje seznamy programovacích technik, které zjednodušují programování Dialogic hlasové knihovny.
- Dodatek A obsahuje seznam vstupů a výstupů hlasových zařízení, pro standardní provozní knihovnu (SRL, Standard Runtime Library).
- Dodatek B definuje seznam chyb hlasové knihovny.
- Dodatek C popisuje rozdíly a podobnosti mezi DTMF a MF tóny.
- Dodatek D poskytuje seznam souvisejících publikací Dialogic.
- Rejstřík a index je také přiložen.

1. 2. Popis funkcí Voice Library

Po konzultaci s vedoucím bakalářské práce bylo rozhodnuto, že aplikace pro ovládání telefonní karty bude v asynchronním režimu. Proto se tato část věnuje pouze popisu funkcí v asynchronním režimu.

Funkce ovládání zařízení

<code>dx_close()</code>	· uzavře desku nebo kanál
<code>dx_open()</code>	· otevře desku nebo kanál

Konfigurační funkce

<code>dx_clrdigbuf()</code>	· vymaže firmwarový zásobník číslic
<code>dx_getparm()</code>	· získá parametr zařízení desky/kanálu
<code>dx_setdigtyp()</code>	· nastaví typ souboru číslic
<code>dx_sethook()</code>	· nastaví stav spínače sluchátka
<code>dx_setparm()</code>	· nastaví parametry zařízení
<code>dx_wtring()</code>	· čeká na počet zvonění

Vstupně/výstupní funkce

<code>dx_dial()</code>	· vytočí ASCII znak příslušné číslice
<code>dx_getdig()</code>	· načte čísla z kanálu digitálního zásobníku
<code>dx_play()</code>	· přehraje hlasová data z jednoho nebo více zdrojů
<code>dx_playiottdata()</code>	· přehraje data z vícenásobných zdrojů
<code>dx_rec()</code>	· nahraje hlasová data do jednoho nebo více cílů
<code>dx_reciottdata()</code>	· nahraje hlasová data do vícenásobných cílů
<code>dx_setdigbuf()</code>	· nastaví režim zásobníku číslic
<code>dx_stopch()</code>	· zastaví aktuální vstup/výstup

Rozšířené funkce

<code>dx_playf()</code>	· přehraje hlasová data z jednoho souboru
<code>dx_playwav()</code>	· přehraje soubor WAVE
<code>dx_playvox()</code>	· přehraje soubor VOX

dx_recf()	·	nahraje hlasová data do jednoho souboru
dx_recwav()	·	nahraje hlasová data do jednoho WAVE souboru
dx_recvox()	·	nahraje hlasová data do jednoho VOX souboru

Funkce příchodu událostí o stavu hovoru

dx_getevt()	·	získá událost stavu hovoru
dx_setevtmsk()	·	nastaví oznámení události stavu hovoru

Pro použití funkcí je nutné zahrnout hlavičkové soubory `srllib.h` a `dxxlib.h`. Návrátová hodnota „0“ značí úspěšnou inicializaci, pro ošetření chyby slouží návratová hodnota „-1“.

Hodnoty událostí

TDX_PLAY	0x81	přehrávání ukončeno
TDX_RECORD	0x82	nahrávání ukončeno
TDX_GETDIG	0x83	zadání číslic ukončeno
TDX_DIAL	0x84	vytáčení ukončeno
TDX_CALLP	0x85	proces hovoru ukončen
TDX_CST	0x86	událost CST odeslána
TDX_SETHOOK	0x87	nastavení sluchátka ukončeno
TDX_WINK	0x88	pokyn odeslán (wink completed)
TDX_ERROR	0x89	událost chyby
TDX_PLAYTONE	0x8A	přehrání tónu ukončeno
TDX_GETR2MF	0x8B	R2MF signály ukončeny
TDX_BARGEIN (Barge	0x8C	ukončení neočekávaného vstupu do hovoru in completed)
TDX_NOSTOP	0x8D	ukončení nemusí být vydáno
TDX_TXDATA	0x90	přenos dat ukončen
TDX_RXDATA	0x91	přenos dat ukončen

Poznámka: CST (call status transition) = změna statusu volání

1.3. Asynchronní a synchronní režim

Asynchronní programování umožňuje ovládání jedním programem vícenásobné hlasové kanály uvnitř jednoho vlákna (thread). Tím je dovolen vývoj komplexních aplikací tam, kde musí být vícenásobné úlohy koordinovány současně. Asynchronní programovací model podporuje volání událostí systému Windows a správu volání událostí.

Synchronní programovací model používá funkce, které blokují běh aplikací po dobu vykonávání funkce. Tento model vyžaduje, aby aplikace ovládala každý kanál z odděleného vlákna nebo procesu. Model umožňuje přiřazení různým aplikacím k různým kanálům v reálném čase.

Rozšířený asynchronní programovací model je podobný klasickému asynchronnímu, kromě implementace použití funkce `sr_waitvtEx()`. To dovoluje aplikaci použití různých vláken čekajících na události od různých zařízení. Stejně jako u základního asynchronního modelu mohou být události spuštěny z různých vláken a dokončeny až po vyzvednutí události z vlákna pomocí `sr_waitvtEx()`.

Po konzultaci s vedoucím bakalářské práce bylo přistoupeno k řešení pomocí rozšířeného asynchronního modelu. K zachytávání událostí bylo vytvořeno vlákno, které zachytává jen události týkající se karty D/4PCI. Navíc bylo možné pomocí funkcí k maskování událostí nastavit zachytávání pouze těch událostí, které byly nezbytně nutné k vykonávání dané činnosti.

Příklad nastavení masky událostí v rozšířeném asynchronním režimu:

```
if(dx_setevtmsk(myPr->dev, DM_RINGS | DM_RNGOFF | DM_DIGITS | DM_DIGOFF) == -1)
```

Tento řádek nastavuje čekání na vyzvánění, čekání na zavěšení a vypne hlášení stisknutých číslic. Celá syntaxe je jako argument příkazu `if`, kde se porovnává s hodnotou „-1“. Jak je již výše napsáno, jedná se o návratovou hodnotu většiny funkcí z Voice API knihovny reprezentující chybu. Za příkazem `if` by tedy mohlo následovat ošetření chyby při nastavování masky, např. `dialog`.

Po spuštění programu tedy dochází k inicializaci vlákna. Současně s tím se nastaví maska událostí. Pokud bude nutno během chodu programu některou událost

„odfiltrvat“, lze nastavení masky kdykoliv změnit. Poté probíhá vlákno cyklicky při příchodu jakékoliv události a testuje její příznak.

Příklad testování události:

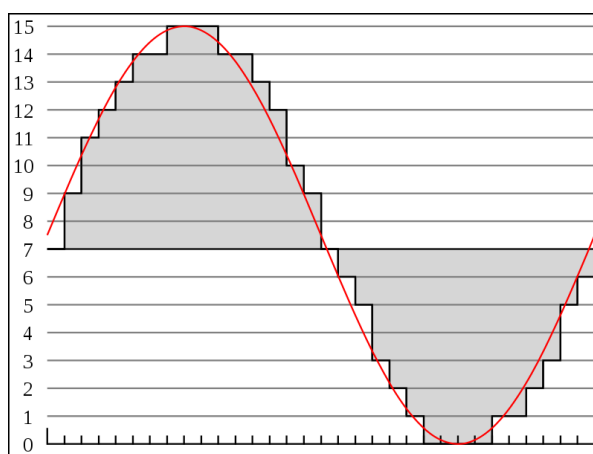
```
if(evt == TDX_PLAY) //buffer byl prehran
{
    myPr->status = TDX_PLAY;
}
```

Na událost TDX_PLAY zareaguje vlákno pouze nastavením proměnné status na hodnotu TDX_PLAY, tedy 0x81h. Tímto způsobem je možné odfiltrvat události určené pouze pro kartu Dialogic a ihned je zpracovávat, nebo dále předávat.

2. Formáty pro záznam zvuku

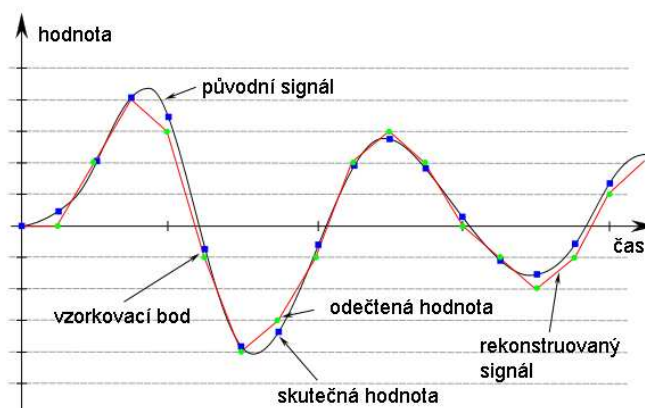
2.1. Pulzně kódová modulace PCM

Pulzně kódovou modulaci vytvořil Brit Alec Reeves již v roce 1937 a dodnes se jedná o jeden z nejpoužívanějších způsobů digitalizace analogového signálu. Základní princip spočívá v odečtení hodnoty analogového signálu pomocí analogově-digitálního převodníku v pravidelném intervalu, tedy s určitou vzorkovací frekvencí. Právě ta je ještě společně s jemností rozlišení převodníku hlavním faktorem určující „věrohodnost“ výsledného signálu.



obr.1: Ukázka digitalizace analogového signálu pomocí čtyřbitové pulzně kódové modulace (zdroj wikipedia.cz)

Vzorkovací frekvence se pohybují v hodnotách od 8kHz a její hodnota by vždy měla být minimálně dvojnásobkem frekvence zaznamenávaného signálu. Jinak vzniká jev zvaný aliasing (signály si neodpovídají amplitudou ani frekvencí), který výrazně znehodnotí výstupní signál. Rozlišení je dáno použitým typem převodníku a udává s jakou přesností jsou kvantovány jednotlivé hodnoty. Nejčastěji se používá osm , dvanáct nebo šestnáct bitů k určení hodnoty signálu. Rozlišení je přímo úměrné velikosti kvantizačního šumu. Velikost šum je definovaná jako rozdíl skutečné a odečtené hodnoty.



obr.2: Šum vzniklý rozdílem skutečné a odečtené hodnoty(zdroj wikipedia.cz)

Pulzně kódová modulace se dnes běžně využívá například v digitálních telefonních linkách ISDN (se vzorkovací frekvencí přes 44,1 kHz). Audio CD ve standartu Red Book používá na dvou kanálech (stereo) 16 bitů s frekvencí 44,1 kHz. Jedna minuta záznamu tedy zabírá více než 10 MB, což je důvod používání formátů pro záznam zvuku s využitím ztrátové komprese. Tím jsou získána data se zhruba desetkrát menší velikostí při téměř nezměněné subjektivní kvalitě signálu. V telekomunikační technice se PCM označení dále nerozvádí a je tak označeno 8 bitů se vzorkovací frekvencí 8kHz (podle mezinárodního standartu ITU G.711). Tím je daný základní datový tok ISDN linky, což je 64 kbit/s. Takový hovorový kanál bývá někdy označován jako DS0 nebo E0.

2.2. Logaritmická komprese A-law, μ -law

Linky DS0 (digital signal 0, americký standart) a E0 (evropský ekvivalent DS0) používají pro kódování logaritmickou kompresi. V Evropě a Austrálii je tento způsob známý pod označením A-law, v Americe a Japonsku jako μ -law. Dochází k převodu dvanácti či třináctibitového signálu na osmibitový (A-law) nebo sedmibitový (μ -law). Linka DS0 totiž používá pro přenos informace pouze 7 bitů. Osmý bit se využívá k signalizaci. Standart E0 používá všech 8 bitů pro přenos signálu a signalizace je samostatně. V evropském standartu je tedy hovor „kvalitnější“, nežli například v Japonsku. Při přechodu z jedné sítě do druhé musí dojít k překódování. Informaci o použitém kodeku nese právě zmíněný bit signalizace.

2.3. Kódování DPCM, ADPCM

Jedná se o kompresní algoritmy původně vytvořené firmou Intel. Zkratka ADPCM znamená Adaptive Delta Pulse Code Modulation (adaptivní delta pulzní kódová modulace). Na rozdíl od klasického PCM je každý vzorek zaznamenán jako odchylka (delta) od předchozího vzorku, což snižuje bitrate. Při vzorkovací frekvenci je potřeba na kompresi pomocí μ -law PCM 64 kb/s, pro ADPCM pouze 32 kb/s. Rozdíl mezi ADPCM a DPCM je pouze v tom, že ADPCM je exponenciální, zatímco DPCM je lineární, což zvyšuje rychlost přeběhu. Oba formáty se dnes také používají ve sdělovací technice.

2.4. Použití v dialogovém systému

Při realizaci dialogového systému bylo nutné použít jak logaritmickou kompresi a-law tak lineární PCM. Telefonní karta Dialogit D4/PCI používá standardně formát A-law, zatímco algoritmy rozpoznávače a syntetizátoru používají PCM. Pro konverzi a zpětnou konverzi byly vytvořeny funkce s názvem `linear2ulaw()` a `ulaw2linear()`. K realizaci se nabízely dvě základní možnosti, přepočítávání hodnot nebo přiřazení hodnoty podle předem přepočítané tabulky. V tomto případě se jedná o převod osmi bitů, takže bylo rozhodnuto pro druhou variantu.

Ukázka kódu funkce ulaw2linear():

```
short Ckarta_D4::ulaw2linear(unsigned char ulawbyte)
{
    static short linear [256]={
0 , -31611 , -30587 , -29563 , -28539 , -27515 , -26491 , -25467 , -24443 , -
23419 , -22395 , -21371 , -20347 , -19323 , -18299 , -17275 , -16251 , -15739
, -15227 , -14715 , -14203 , -13691 , -13179 , -12667 , -12155 , -11643 , -
11131 , -10619 , -10107 , -9595 , -9083 , -8571 , -8059 , -7803 , -7547 , -7291
, -7035 , -6779 , -6523 , -6267 , -6011 , -5755 , -5499 , -5243 , -4987 , -4731
, -4475 , -4219 , -3963 , -3835 , -3707 , -3579 , -3451 , -3323 , -3195 , -3067
, -2939 , -2811 , -2683 , -2555 , -2427 , -2299 , -2171 , -2043 , -1915 , -1851
, -1787 , -1723 , -1659 , -1595 , -1531 , -1467 , -1403 , -1339 , -1275 , -1211
, -1147 , -1083 , -1019 , -955 , -891 , -859 , -827 , -795 , -763 , -731 , -699
, -667 , -635 , -603 , -571 , -539 , -507 , -475 , -443 , -411 , -379 , -363 , -
347 , -331 , -315 , -299 , -283 , -267 , -251 , -235 , -219 , -203 , -187 , -171
, -155 , -139 , -123 , -115 , -107 , -99 , -91 , -83 , -75 , -67 , -59 , -51 , -43
, -35 , -27 , -19 , -11 , -3 , 31612 , 30588 , 29564 , 28540 , 27516 ,
26492 , 25468 , 24444 , 23420 , 22396 , 21372 , 20348 , 19324 , 18300
, 17276 , 16252 , 15740 , 15228 , 14716 , 14204 , 13692 , 13180 ,
12668 , 12156 , 11644 , 11132 , 10620 , 10108 , 9596 , 9084 , 8572 ,
8060 , 7804 , 7548 , 7292 , 7036 , 6780 , 6524 , 6268 , 6012 , 5756 ,
5500 , 5244 , 4988 , 4732 , 4476 , 4220 , 3964 , 3836 , 3708 , 3580 ,
3452 , 3324 , 3196 , 3068 , 2940 , 2812 , 2684 , 2556 , 2428 , 2300 ,
2172 , 2044 , 1916 , 1852 , 1788 , 1724 , 1660 , 1596 , 1532 , 1468 ,
1404 , 1340 , 1276 , 1212 , 1148 , 1084 , 1020 , 956 , 892 , 860 , 828
, 796 , 764 , 732 , 700 , 668 , 636 , 604 , 572 , 540 , 508 , 476 ,
444 , 412 , 380 , 364 , 348 , 332 , 316 , 300 , 284 , 268 , 252 , 236
, 220 , 204 , 188 , 172 , 156 , 140 , 124 , 116 , 108 , 100 , 92 , 84
, 76 , 68 , 60 , 52 , 44 , 36 , 28 , 20 , 12 , 4 , 0 , };
return(linear[ulawbyte]);}
```

Funkce se volá s jediným parametrem, což je hodnota určitého bitu ve formátu A-law. Výstupem je odpovídající hodnota PCM. Tím, že funkce převádí bit po bitu, je možné jí pomocí for cyklu použít k převodu libovolně dlouhého signálu.

Volání funkce ulaw2linear() pro převod dvou sekund signálu (16 000 vzorků):

```
char x[16000]; long i;
for(i=0; i<16000; i++)
    x[i]=ulaw2linear(rec1[i]);
```

Tento cyklus provede převod hodnot A-law v proměnné rec1, deklarované jako pole charů, na hodnoty PCM a uloží je do pole charů s názvem x. Příklad je pouze ilustrační a je byl takto použitelný pouze při ladění aplikace, kdy bylo známo, že počet vzorků je 16 000 (odpovídá dvěma sekundám záznamu). Při použití během dialogu bude možné testovat počet nenulových vzorků a podle toho alokovat velikost pole x a upravit počet cyklů. Zkrátí se tím doba potřebná pro převod, což může pomoci při optimalizaci reakční doby systému. Samozřejmostí je i funkce pro zpětný převod

nazvané `linear2ulaw()`. Její volání a použití je shodné s funkcí `ulaw2linear()`. Více o zvukových formátech v [2].

Ukázka kódu funkce `linear2ulaw()` pro zpětnou konverzi:

```
unsigned char Ckarta_D4::linear2ulaw(short sample)
{static short exp_lut[256] = {0,0,1,1,2,2,2,2,3,3,3,3,3,3,3,3,
                               4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,
                               5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,
                               5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,
                               6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,
                               6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,
                               6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,
                               6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,
                               7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,
                               7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,
                               7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,
                               7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,
                               7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,
                               7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,
                               7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,
                               7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7};

    short sign, exponent, mantissa;
    unsigned char ulawbyte;
    sign = (sample >> 8) & 0x80;
    if (sign != 0) sample = -sample;
    if (sample > 32635) sample = 32635;
    sample = sample + 0x84;
    exponent = exp_lut[(sample>>7) & 0xFF];
    mantissa = (sample >> (exponent+3)) & 0x0F;
    ulawbyte = ~(sign | (exponent << 4) | mantissa);
    return(ulawbyte);}
```

3. Distribuovaný hlasový rozpoznávač

3.1. Základní princip

Hlavním důvodem, proč se přistupuje k rozpoznávání řeči v distribuované podobě je samozřejmě snížení hardwarových nároků na provoz daného systému, jednoduchá modifikovatelnost a v neposlední řadě také ochrana zdrojového kódu rozpoznávače, který obsahuje dlouhodobě shromažďovaná a velmi cenná data. Vstupní zařízení provede záznam hlasu, parametrizaci daného signálu a pomocí sítě internet zasílá data na server, kde dojde k rozpoznání a zpětného zaslání výsledku procesu (slovo v textové podobě). Systém má tři základní úkoly k úspěšném rozpoznání daného signálu:

- Parametrizace signálu
- Detekce řečové aktivity
- Klasifikace

Parametrizace je proces transformace PCM vzorků na posloupnost vektorů rozpoznávacích příznaků, tedy uspořádaných n -tic které reprezentují obsah nahrávky. Pro potřeby rozpoznávání je potřeba dostatečná kvalita signálu, která je většinou dána vzorkovací frekvencí. Minimum pro kvalitní interpretaci řeči pro rozpoznávač je 8kHz. Nejčastějším způsobem reprezentace řečového signálu je keprální analýza využívající transformaci spektra krátkého úseku signálu k oddělení informace o buzení hlasového traktu a informace o odezvě hlasového traktu.

Detekce řečové aktivity je proces zajišťující oddělení jednotlivých slov, resp. vyhledávání pomlek mezi nimi. Detekuje se nárůst energie slova na jeho začátku nebo pokles energie na jeho konci. Pokud však chceme detekovat pomlky i v případech, kdy je na pozadí parazitní hluk či šum, je nutné použít systémy detekce založené na detekci charakteristických rysů lidského hlasu využívající keprální analýzu.

Při klasifikaci je potřeba zpracovaný signál (resp. sadu příznaků, která ho charakterizuje) porovnat s databází vzorů. Kvůli různé rychlosti promluvy mluvčích se používají pro komparaci vzorků metody pro měření podobnosti různě dlouhých řad.

Databáze vzorů musí být nezávislá na pohlaví a věku mluvčího. Proto se z pravidla vytváří na základě realizací daného slova od velké skupiny mluvčích. Pro pozdější možnost přidávání slov do slovníku není potřeba nevytvářet databázi celých slov, jako databázi základních stavebních prvků slov jako jsou hlásky, nebo jejich dvojice nebo trojice, ze kterých je možné sestavení daného vzoru pouhým zřetěžením.

Více o rozpoznávání hlasu v [6].

3.2. Použitý rozpoznávač izolovaných slov

Rozpoznávací systém využívaný v této aplikaci vytvořil Ústav informačních technologií Technické univerzity v Liberci. Tento systém umožňuje rozpoznávání významu izolovaných slov a krátkých větních spojení. Má mnoho užitečných funkcí, jako je potlačení šumu a hluku. Hlavní výhodou je v možnosti použití dvou slovníků. První slovník je implementován přímo v systému na serveru a není možné do něho zasahovat. Jeho kapacita je cca několik tisíc slov. Druhý slovník je uložen na pevném disku počítače DOK02 na kterém běží aplikace dialogového systému a může obsahovat až dva tisíce slov.

Komunikace klienta se serverem probíhá pomocí sítě v reálném čase. Proto je celý algoritmus koncipován s co nejnižším nárokem na datové toky. Pro parametrizaci je použita lineární asynchronní metoda, která pracuje nezávisle na vstupním signálu. Rámec je dlouhý 25 ms a sousedící rámce se překrývají o 10 ms. Asynchronní metoda totiž nerozděluje rámce podle signálu na celé periody, nýbrž na pevnou délku rámce, takže je nutné pro před použitím FFT vynásobit rámec Hammingovým okénkem. Tím dochází ke ztrátě dat na začátku a na konci každého okénka, což je kompenzováno 10ms překrytím. Samotná parametrizace se provádí keprstrální analýzou MFCC pomocí 13 keprstrálních koeficientů, delta příznaků a deltadelta příznaků (první a druhá derivace). Pro detekci začátku a konce promluvy je použit logaritmus energie rámce. Detektor řeči sledující energii signálu se dokáže adaptovat na úroveň šumu na pozadí v neřečových oblastech.

3.3. Implementace rozpoznávače

Vedoucí bakalářské práce poskytl funkční klientskou aplikaci v prostředí C++, která byla implementována do systému. Aplikace zajišťuje připojení a odpojení rozpoznávače, inicializaci rozpoznávacího slovníku a spuštění (ukončení) rozpoznávání.

Po připojení dojde k odeslání slovníku (seznam telefonních čísel zaměstnanců Ústavu informačních technologií ITE včetně modifikací výslovnosti) na server, kde dojde k vytvoření modelu. Po přijmutí příchozího hovoru a přehrání úvodního monologu dojde k čekání na odezvu volajícího. Nahrávka částí hovoru se odesílá pomocí sítě internet z počítače DOK02 do rozpoznávacího modulu. Zde je promluva parametrizována a je vyhodnoceno rozpoznání pomocí nejbližší shody se vzorem ze slovníku rozpoznávače. Reakcí je odpověď podle uživatelského slovníku. V tomto případě se jedná o jméno, příjmení (implicitní potvrzení) a telefonní číslo, resp. klapku na ústředně Technické univerzity v Liberci (poslední čtyřčíslí).

Více o distribuovaném rozpoznávači v [7].

4. Syntéza řeči

Základní způsoby syntézy lidské řeči bychom mohli rozdělovat podle kvality a věrohodnosti výsledného signálu, stejně jako podle náročnosti na provedené požadovaných algoritmů. Náplní této práce je však pouze seznámení s možnostmi řečové syntézy a zejména jejich implementace do telefonního dialogového systému. Základní metody hlasové syntézy jsou následující:

- Lineární model hlasu
 - formantový model
 - LPC model
 - kepsrální model
- Sinusový model
- Spojování řečových úseků – PSOLA

4.1. Lineární modely

Lineární model řeči využívá linearizaci řečového signálu, jak napovídá již název. Využívá ho formantová, kepsrální a LPC syntéza. Lidský řečový trakt totiž není kvůli své stavbě schopen rychlých změn. Na krátkých úsecích tedy můžeme hlasový signál prohlásit za stacionární aniž bychom se u většiny hlásek dopouštěli závažnější chyby. Délka linearity se pohybuje v rozmezí 10 až 30 ms, u dětského hlasu je nepatrně kratší. Rozdělení hlasového signálu na segmenty (framy) můžeme provádět buď synchronní nebo asynchronní metodou. Synchronní metoda rozděluje řečový signál přesně po dvou až třech periodách. Asynchronní metoda rozděluje po přesných časových úsecích bez ohledu na periodu vstupního signálu. Tím dochází k tomu, že u asynchronní metody není vzorek přesně periodický, takže pro zpracování pomocí FFT je nutné ho vynásobit okénkovou funkcí (nejčastěji Hammingovo okénko), čímž dochází ke ztrátě informací na začátku a konci periody.

Lineární model se skládá ze dvou hlavních částí, buzení (znělé nebo neznělé) a číslicový filtr. Buzení modeluje proud vzduchu hlasivkami a filtr modeluje hlasový trakt.⁴

4.1.1. Formantový model

Formantové syntetizátoru modulují výsledný signál podobně jako v lidském hlasovém traktu pomocí rezonátorů a antirezonátorů. Tím je možné ovlivňovat jednotlivé parametry hlasu zvlášť, což odpovídá i výsledné kvalitě signálu. Skládá se ze zdroje buzení (znělého a neznělého) a z posloupnosti rezonátorů a antirezonátorů. Rezonátory slouží k modelování spektrálních vrcholů (formantů) a specifikují se pomocí rezonanční frekvence a šířky pásma. Realizace se provádí pomocí zpětnovazebního filtru druhého řádu. Antirezonátory modelují vliv vedlejších rezonátorů v hlasovém traktu (vytvářejí antiformanty) a k simulaci hlásek které vznikají v dutině nosní. Nejpoužívanější je sériově/paralelní model vyvinutý D. Klattem.

4.1.2. LPC model

Pomocí lineárního prediktivního kódování (*Linear Predictive Coding*) model odhaduje přenosovou funkci z výsledků analýzy signálu. Modelování řeči pomocí LPC koeficientů dosahuje lepších parametrů než formantová metoda, která modeluje hlasový trakt jako systém několik nezávislých filtrů, zatímco LPC je realizována jako jeden IIR filtr.

4.1.3. Kepstrální model

Tento model podobně jako formantový simuluje jak formanty, tak antiformanty. K tomu model využívá podobně jako LPC přechodovou funkci. Pomocí aproximace kepstrálních parametrů lze však dosáhnout modelu s libovolnou přesností a odstranit tak například chybu lineární predikce, která vzniká u některých samohlásek.

4.2. Sinusový model

Sinusový model získává parametry signálu pomocí FFT z jeho frekvenční oblasti a to výběrem lokálních maxim amplitudového spektra nebo aproximací obálky amplitudového spektra.

4.3. Syntéza skládáním řečových úseků, PSOLA

Nejstarší systém pro syntézu řeči vytvářel výstupní signál složením z nahraných úseků řeči. Nahraná řeč je rozdělena na jednotky o určené délce, které jsou uloženy v paměti syntetizátoru. Vhodným spojováním pak dochází k syntéze. V případě některých telefonních dialogových systému je dokonce nahrána celá promluva, což vede k velmi kvalitní interpretaci za cenu nemožnosti jakékoliv změny. Modifikací této metody vznikla metoda PSOLA, která dokáže velmi kvalitní syntézu hlasu, ale při modelování prozodie je velmi omezena a dochází ke snížení kvality.

4.4. Použitý modul syntézy - Modul TTS

Jedná se o dynamickou DLL knihovnu připojenou k dialogovému systému, kterou poskytl již ve funkční podobě vedoucí bakalářské práce Ing. Holada. Modul se chová jako klient, takže se jedná o distribuovanou syntézu. Systém se jmenuje EPOS a byl vyvinut na akademii věd Ing. P. Horákem Ph.D.

5. Návrh telefonního dialogového systému

5.1. Typ dialogového systému

Ze zadání této práce vyplývá několik vlastností, které je třeba kategorizovat nebo upřesnit. Systém vede s volajícím dialog s konečným počtem stavů s iniciativou systému s některými prvky smíšené iniciativy. Díky jednoduchosti dialogu je možné použít implicitní potvrzení, tedy potvrzení zapojení přímo do odpovědi na daný dotaz. Vedení dialogu je koncipováno pro cílovou skupinu dospělí s mírně pokročilými schopnostmi a se zkušeností s jinými dialogovými systémy. K samotnému rozpoznávání je využit již fungující rozpoznávač izolovaných slov v distribuované podobě.

5.2. Diagram telefonního dialogového systému

Pro snazší pochopení struktury tohoto systému byl vytvořen diagram (na obr.), který znázorňuje základní bloky programu a vazby mezi nimi. Prostřední větev diagramu znázorňuje ovládání systému, levá větev nahrávání hovoru a zpracování nahrávky, pravá větev reprezentuje tvorbu hlasové odpovědi systému.

Základním blok „Telefonní karta Dialogic“ je umístěn v nejvyšší části diagramu a je tím myšlen samotný hardware. Telefonní karta je řízena základním vláknem (thread), do kterého posílá události a přijímá z něho příkazy. Kvůli možnosti maskování (filtrování) událostí bylo vytvořeno druhé pomocné vlákno, do kterého přicházejí jen maskou určené události. Zde dochází také k jejich zachytávání a zpracování dále do systému, nebo v některých případech rovnou ke zpracování příkazu (vytvoření reakce na událost).

rozpoznávače izolovaných slov. Po rozpoznání dojde k zaslání významu slov v textové podobě.

V tuto chvíli přeruší nahrávání pouze pokud je v obsahu zásobníku detekován zavěšovací tón. Algoritmus pro jeho detekci vyšle příkaz do vlákna aplikace, která hovor ukončí nastavením hookstate pomocí `dx_sethook()` na hodnotu `DX_ONHOOK`.

Ukázka „položení sluchátka“:

```
dx_sethook(myPr->dev, DX_ONHOOK, EV_ASYNC);
```

Pokud je slovo či fráze z rozpoznávače ve slovníku, tedy rozpoznávač zná reakci na uvedenou otázku, zastaví nahrávání a vybere ze slovníku příslušnou odpověď, kterou pošle v textové podobě zpět. Systém ji předává do syntetizátoru (pomocí modulu Text To Speech). Následuje odpověď systému ve zvukové podobě, která je nahrána do výstupního zásobníku ze kterého se přehrává přes telefonní kartu volajícímu.

5.3. Test na zavěšení

Na první pohled by se mohlo zdát, že testování na zavěšení protistrany pomocí detekce zavěšovacího tónu takto přímo v aplikaci není úplně ideálním řešením. Bohužel právě tato problematika zabrala při konečné realizaci systému nejvíce času. Telefonní karta má samozřejmě podporu detekce DTMF tónů, detekci ticha, dokonce i detekci pulzní volby. V asynchronním režimu bohužel nelze standardně využít všech možností. Ve vláknu aplikace byly i při „úplném odmaskování“ zachytávány různé události, ale událost `TDX_CALLP (0x85h)` která je posílána jako reakce na zavěšovací tón nebyla během testování nikdy zachycena.

Důsledek absence této události byl takový, že po ukončení hovoru protistranou zůstala linka obsazena a nebyl možný další příchozí hovor. Muselo dojít k „manuálnímu zavěšení“ pomocí tlačítka na testovacím rozhraní aplikace. Z tohoto důvodu bylo po necelých dvou měsících řešení této problematiky odstoupeno od ukončování hovoru standardním způsobem a bylo naimplementováno již funkční (i když ne příliš elegantní) řešení s detekcí zavěšovacího tónu přímo v aplikaci.

5.4. Slovník

Slovník je jednoduchý textový soubor s příponou „*.vcb“, ve kterém jsou data uspořádána podle vzoru otázka – odpověď. Pro jednodušší použití je každá položka (myšleno odpověď, tedy telefonní číslo) napsána hned několikrát. Pokaždé však jako reakce na jinou otázku. Tím jsou velmi jednoduše vyřešeny možnosti výslovnosti různých jmen.

Ukázka zápisu jména Ing. Jan Silovský:

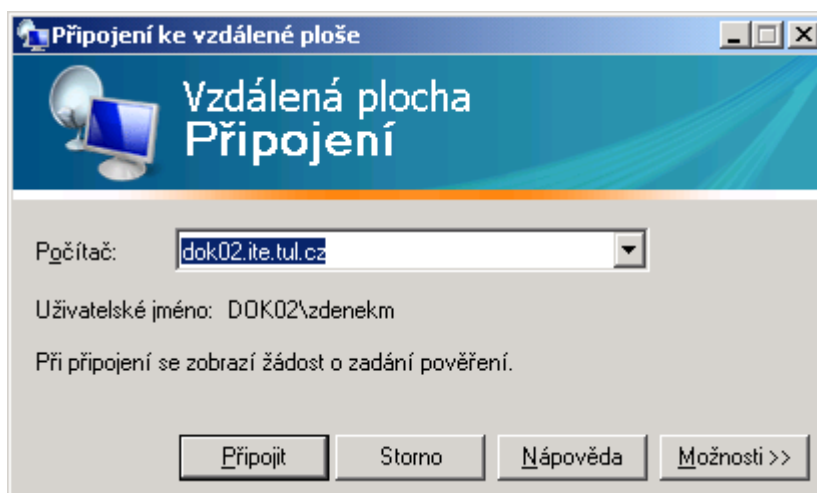
"Jan Silovský, 3778" "jan silovský"
"Jan Silovský, 3778" "honza silovský"
"Jan Silovský, 3778" "jan silovskej"
"Jan Silovský, 3778" "silovský"
"Jan Silovský, 3778" "honza silovskej"
"Jan Silovský, 3778" "inženýr jak silovský"
"Jan Silovský, 3778" "inženýr silovský"

Na ukázce je vidět že není počítáno se všemi možnými variantami. Například „Honza Silovskej“ by mohl říct mluvčí, který má osobní vztah s danou osobou. Pokud použije daný nespisovný tvar, určitě nepřidá k oslovení také titul. Naopak mluvčí který osobu nezná zřejmě použije oslovení včetně titulu. Titul bude vždy na prvním místě, poté bude následovat jméno a příjmení, nebo jen příjmení. Formulace ve tvaru titul, příjmení a jméno se nepoužívá, takže není třeba ji zařazovat.

6. Testování systému

6.1. Připojení ke vzdálené ploše

Poměrně nestandardní byl již vývoj aplikace. Kvůli nutnosti použití telefonní karty byl poskytnut ke konečné realizaci osobní počítač DOK 02 z ústavu ITE, který je tímto hardwarem vybaven. Počítač se fyzicky nachází v prostorech, které nejsou veřejně přístupné, proto byl zřízen účet a vývoj aplikace byl prováděn pomocí aplikace MS Windows Připojení ke vzdálené ploše. Tento způsob se velmi osvědčil a do budoucna bude rozumné ho využít k realizaci dalších projektů. Odstraňuje se tak problém s hardwarovým, nebo i softwarovým vybavením studentů, kteří často musí pracovat s lite verzí softwaru potřebného k vypracování projektu.



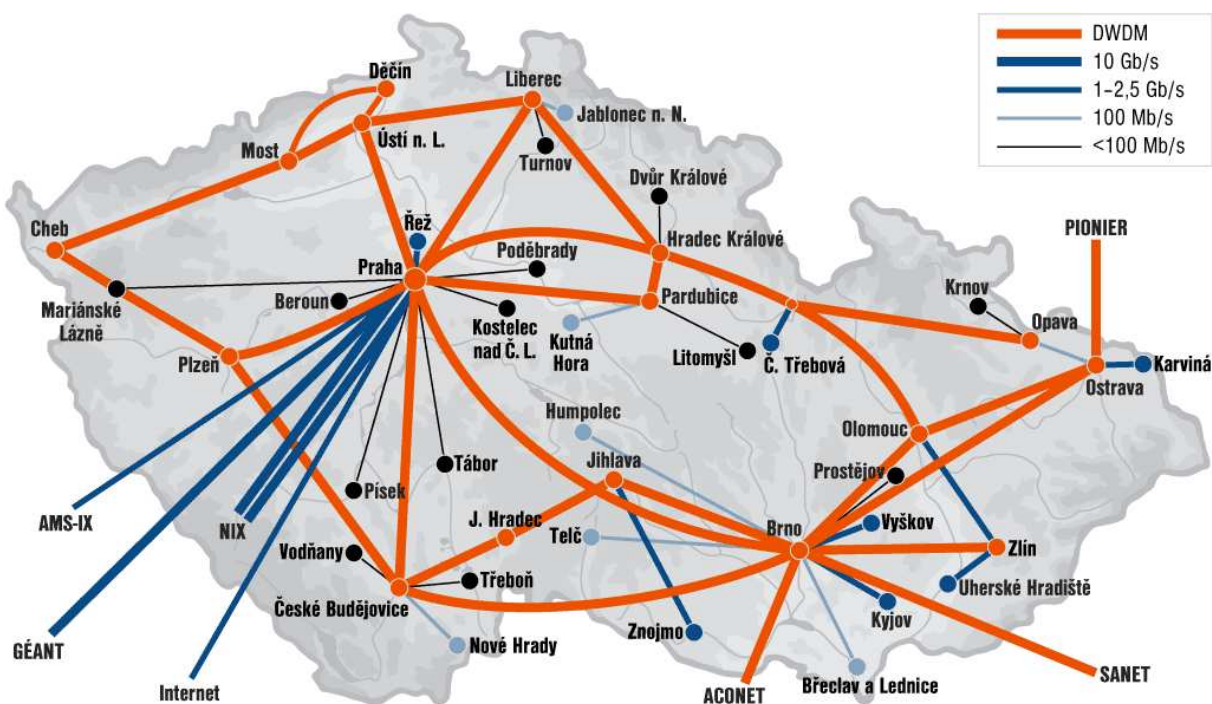
Obr 4.: Připojení ke stanici DOK02

K ladění bylo potřeba provést několik set hovorů na linku 3100. Nejprve bylo ladění prováděno pouze v kabinetu vedoucího bakalářské práce, Ing. Holady. Pro možnost neomezeného testování byl vytvořen účet v síti CESNET2. Podrobný popis je na stránkách <https://sip.cesnet.cz/dokuwiki/>, heslo je možné vytvořit na adrese <https://shibbo.tul.cz/idp/Authn/UserPassword>. Pomocí SIP protokolu bylo možné volat na linku 3100 pomocí klienta X-lite 3.0. Tímto způsobem bylo možné ladění telefonního systému s minimálními náklady. Navíc se tím objevily možnosti pro další rozšíření systému.

6.2. SIP telefonie v síti CESNET2

6.2.1. Síť CESNET2

Tato síť slouží pro vědu, výzkum, vývoj a vzdělání. Propojuje vysokorychlostními okruhy většinu velkých měst v ČR, resp. většinu velkých institucí, jako jsou vysoké školy, Akademie věd České republiky, nemocnice, knihovny nebo i některé střední školy. Kromě velkých přenosových kapacit pro vědecké účely a standardního vysokorychlostního připojení k internetu nabízí síť uživatelům i videokonference nebo IP telefonii.



Obr.5 : Topologie síť CESNET2 (zdroj www.cesnet.cz)

6.2.2. VoIP telefonie

Voice over IP je souhrnné označení technologie pro přenos hlasu pomocí počítačových sítí. Jedná se o jedno z aktuálních témat současnosti, například kvůli efektivnějšímu využití přenosových cest. V klasických telefonních sítích totiž nejsou data na rozdíl od počítačových sítí nijak komprimována a pásmo je staticky přiděleno i

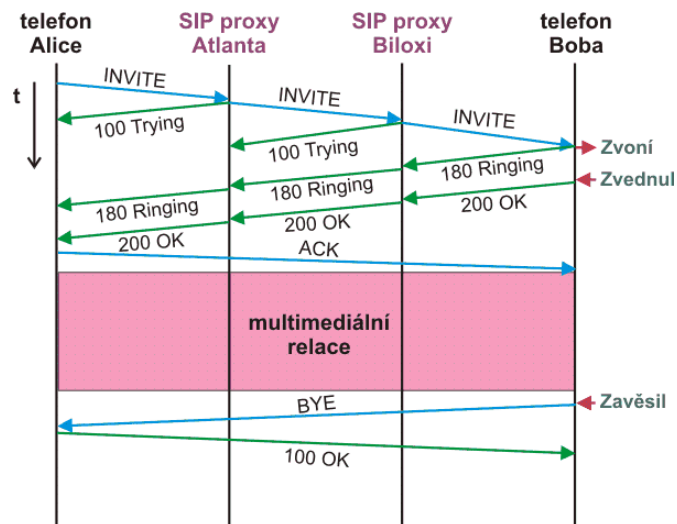
v případech, kdy nikdo nehovoří. Přenos hlasu v počítačových sítích využívá APM sítě (pak hovoříme o tzv. APM telefonii) kvůli parametrům jako garantované parametry spojení, nebo sítě založené na protokolu IP (IP telefonie).

IP telefonie je aplikace provozovaná na paketově přepínaných sítích s IP protokolem. Využívá algoritmus pro kompresi hlasu tak, že se přenáší jen část spektra a tím až osmkrát zefektivňuje komunikaci. Pro přenos dat v síti IP je potřeba velmi široké pásmo a spolehlivost spojení, naopak hlasové služby vyžadují poměrně konstantní pásmo a garanci doby doručení. Z toho důvodu se pro přenos hlasu i dat prostřednictvím sítě IP implementují technologie ze sítě ATM (Asynchronous Transfer Mode), jako je například QoS (Quality of Service).

VoIP technologie samozřejmě nekončí u spojení typu PC-PC, ale rozšiřuje své možnosti propojením s telefonními sítěmi operátorů. Poté je potřeba zajistit přenos i dalších signalizačních informací. Více o síti cesnet v [4].

6.2.3. SIP protokol

Session Initiation Protocol, neboli protokol pro inicializaci relací je protokol aplikační vrstvy sloužící k přenosům signalizačních informací v sítích IP telefonie. První verzi popisuje dokument RFC 2543 (<http://tools.ietf.org/html/rfc2543>), současnou druhou specifikuje RFC 3261 (<http://tools.ietf.org/html/rfc3261>). Starší ekvivalent tohoto protokolu je standart H.323, který je však velmi složitý. SIP vznikl ve snaze vytvoření co nejjednoduššího protokolu z dlouhou dobu využívaných a prověřených principů HTTP a SMTP a je jim velmi podobný. Spolupracuje s protokolem navázání spojení SDP a protokolem řízení datového toku RTP.



Obr. 6: Transakce protokolu SIP podle standartu RFC 3261 (zdroj wikipedia.cz)

Pro multimediální relaci musí protokol zajistit těchto pět činností:

- **Lokalizace účastníka** – nalezení spojení
- **Zjištění stavu účastníka** – zjištění, zda je možná relace (není obsazeno atd.)
- **Zjištění možností účastníka** – typu kodeku, přenosová rychlost, audio/video HW
- **Navázání spojení** – pomocí protokolů SDP a RTP
- **Řízení spojení** – kontrola změny vlastností během relace, ukončení relace

Více o SIP protokolu v [3].

6.3. X-lite 3.0

Klient pro SIP telefonii je zdarma ke stažení na stránkách výrobce (<http://www.counterpath.net/x-lite.html&active=4>). Nejedná se však o opensource, nýbrž pouze o freeware. K ladění je tato možnost postačující. Po instalaci a spuštění programu se objeví jednoduché grafické rozhraní. „Displej“ zobrazuje chybovou hlášku, protože není připojen k síti. Je nutné nakonfigurovat účet.



Obr. 7: X-lite po prvním spuštění není připojen.

Nejjednodušší způsob je klik pravým tlačítkem na okno aplikace, a výběr položky SIP Account settings... Po chvíli se zobrazí seznam všech účtů SIP. Zatím není žádný nakonfigurován. Nový účet přidáme kliknutím na tlačítko Add... Zobrazí se následující formulář.

The image shows a Windows-style dialog box titled "Properties of Account 1". It has several tabs: "Account", "Voicemail", "Topology", "Presence", and "Advanced". The "Account" tab is selected. Inside, there are two main sections. The first, "User Details", contains five text input fields: "Display Name" (filled with "zdenek.mauer"), "User name" (filled with "zdenek.mauer"), "Password" (filled with ten dots), "Authorization user name" (filled with "zdenek.mauer"), and "Domain" (filled with "tul.cz"). The second section, "Domain Proxy", contains a checked checkbox labeled "Register with domain and receive incoming calls". Below this is a label "Send outbound via:" followed by two radio buttons: "domain" (which is selected) and "proxy" (which is disabled). Next to the "proxy" radio button is a text field labeled "Address" which is empty. At the bottom of the dialog is a "Dialing plan" field containing the text "#1\a\A.T;match=1;prestrip=2;". At the very bottom are three buttons: "OK", "Storno", and "Použít".

Obr. 8: Nastavení účtu SIP

K nastavení účtu stačí vyplnit podle obrázku č. údaje na první záložce. Display name může být uživatelské jméno, nebo libovolné jiné. User name je ve formátu jmeno.prijmeni s tím, že doména (v tomto případě tul.cz) nemusí být zadána. Doplní se automaticky, stejně jako u Authorization user name, které je shodné. Důležité je vyplnit danou doménu, tedy tul.cz a v poli Domain proxy zatrhnout radiobutton domain u položky Send outbound via. Ostatní pole neměníme a potvrdíme všechna dialogová okna. Aplikace se poté začne registrovat do sítě.



Obr. 9: Registrace do sítě CESNET2

Po úspěšné registraci se na displeji zobrazí nápis Ready společně s informací o použitém profilu, resp. uživatelském jméně. Klient je připraven k hovoru.



Obr. 10: Klient X-lite se připojil do sítě a je připraven k hovoru.

6.4. Nahrazení telefonní karty SIP klientem

První inovace která se nabízela bylo samozřejmě nahrazení telefonní karty SIP klientem. Pro volajícího by tato změna nebyla ani pozorovatelná, protože by stále volal na telefonní číslo univerzity, hovor by byl spojován univerzitní ústřednou. Propojení s klientem by bylo možné vytvořit pasivně, nebo aktivně. Pasivní propojení lze provést pouze přes zvukovou kartu, ale k aktivnímu by bylo nutné použít klienta s open source licencí.

6.4.1. Pasivní připojení SIP klienta

Pasivní propojení by bylo možné realizovat pomocí zvukové karty. Telefonní dialogový systém by nepřijímal žádné události a vše by muselo být realizováno detekcí tónů podobným způsobem, jakým je detekován zavěšovací tón ve finální verzi dialogového systému. Řešení by rozhodně nebylo elegantní a znemožňovalo by vytočení telefonního čísla nebo ukončení hovoru, což je funkce potřebná k budoucím rozšiřováním systému.

6.4.2. Aktivní připojení SIP klienta

Výše již bylo zmíněno, že použitý klient X-lite je freeware, nikoliv open source. Za tímto účelem je možné využít software jako Opensip (<http://opensips.org/>). Bude potřeba zajistit zachytávání událostí podobně jako u telefonní karty, což je vhodné téma např. pro diplomovou práci. Finální verze by mohla počítat nikoliv s propojením dialogového systému a klienta, ale třeba i s úplnou implementací klienta do systému.

6.5. Přepojování

Velmi zajímavý způsob rozšíření systému který by znamenal rozšíření funkcí systému a tím i přímo zlepšoval výslednou ergonomii volajícímu je možnost realizace ochozích hovorů, resp. přepojování. Dialog by probíhal stávajícím způsobem a po sdělení žádaného telefonního kontaktu by následoval dotaz, zda si přeje volající přepojit na uvedenou linku. Samotné přepojování pomocí klasického telefonního přístroje probíhá následujícím způsobem:

- krátké zavěšení nebo stisk tlačítka flash
- vytočení nového telefonního

- vyzvánění
- přijmutí hovoru protistranou
- dialog s protistranou
- zavěšení, přepojení

Po zavěšení provede telefonní středa automatické propojení volajícího a volaného – hovor byl přepojen. Dialogový systém by musel mít implementováno vytáčení čísel a ošetřenu možnost, že volaný hovor nepřijme. Dále by musela být vytvořena krátká promluva k volanému, aby věděl že telefonní dialogový systém přepojí jiný hovor. Nejdůležitější ze všeho by však byl hned první bod seznamu, tedy flash, neboli krátké zavěšení. Tento bod by musel být vyřešen delším testováním aplikace, protože tato vlastnost není telefonní kartou standardně podporována.

7. Závěr

Během vývoje aplikace byly splněny všechny zadané úkoly. Zároveň se podařilo rozšířit systém o některé nové funkce. K poskytnuté aplikaci pro komunikaci s distribuovaným rozpoznávačem a k modulu TTS pro distribuovanou syntézu systémem EPOS bylo vyvinuto vhodné prostředí, které je na rozdíl o předchozích systémů jako Infocity kompaktní a přehledné.

Systém nahrávání byl vyřešen použitím dvou zásobníků (bufferů). Při zápisu a zároveň čtení zásobníku by mohlo dojít ke kolizi a tím ke ztrátě části informace. Taková data by rozpoznávač zřejmě nemohl správně identifikovat ani při použití opravných algoritmů. Uvedeným řešením se velmi zkrátila odezva systému. Aplikace pracuje celkově rychle a spolehlivě.

Tento dialogový systém ukazuje možnou budoucnost telefonických hlasových automatů, které známe z infolinek mobilních operátorů nebo bankovních subjektů. Postupně s rozvojem možností hlasových rozpoznávačů bude možné vyvíjet i aplikace, které povedou dialog s opravdu smíšenou iniciativou, nebo dokonce s plnou iniciativou uživatele.

Při nalézání způsobu ladění běhu aplikace a zachytávání událostí vyvolaných příchozím hovorem byl využity možnosti SIP Telefonie pomocí sítě CESNET2. V tu dobu se zrodila myšlenka dalšího rozšiřování současného telefonního dialogového systému.

8. Zdroje

- [1] Vlastní zápisky z přednášek předmětu signály a informace 2008 (přednášející prof. J. Nouza)
- [2] Wikipedia, Pulze kódová modulace [5/2009]
url:(http://cs.wikipedia.org/wiki/Pulzn%C4%9B_k%C3%B3dov%C3%A1_modulace)
- [3] Wikipedia, protokol SIP [5/2009] url:(<http://cs.wikipedia.org/wiki/SIP>)
- [4] Cesnet [5/2009] url:(<http://www.cesnet.cz/iptelefonie/>)
- [5] Martin Pelc, DIPLOMOVÁ PRÁCE, Víceuživatelský hlasový rozpoznávací server s distribuovanými výpočty, Liberec 2006
- [6] Tomáš Klucho, Bakalářská práce, Návrh interaktivního hlasového rozhraní mezi člověkem a počítačem, Liberec 2008
- [7] Petr Koula, DIPLOMOVÁ PRÁCE, Návrh deklarativního jazyka pro hlasové dialogové systémy, Liberec 2008

Příloha A

Technické specifikace karty Dialogic D4/PCI

Počet portů	4
Maximum karet/systémů	16
Mikroprocesor	Intel® 80C188
Digitální signálový procesor	Motorola DSP56002
Sběrnice	PCI (kompatibilní s PCISIG Bus Specification, Rev. 2.1)
Rychlost sběrnice	PCI 33 MHz
Konektory	4 konektory RJ-11
Napájení a zatížitelnost:	
+5 VDC	650 mA
+12 VDC	55 mA
-12 VDC	53 mA
Provozní teplota	0°C až +50°C
Skladovací teplota	-20°C až +70°C
Vlhkost	8% až 80%, nekondenzační
Vstupní rozsah	-50 dBm až -13 dBm (jmenovitý), pro průměrný řečový signál, nastavitelný parametrem
Automatický regulace buzení	Softwarově zapínatelné/ vypínatelné nad -30 dBm v plném rozsahu nahrávání, nastavitelné parametrem
Detekce ticha	-40 dBm jmenovitě, softwarově nastavitelné
Frekvenční citlivost	24 Kb/s 300 Hz až 2600 Hz ± 3 dB 32 Kb/s 300 Hz až 3400 Hz ± 3 dB 48 Kb/s 300 Hz až 2600 Hz ± 3 dB 64 Kb/s 300 Hz až 3400 Hz ± 3 dB
Digitalizace zvuku	24 Kb/s ADPCM @ 6 kHz vzorkování 32 Kb/s ADPCM @ 8 kHz vzorkování 48 Kb/s μ -law PCM @ 6 kHz vzorkování 64 Kb/s μ -law PCM @ 8 kHz vzorkování

Přehrávání wave	Podpora pro 11 kHz lineární PCM, 8-bitový mono režim (dostupné jen pod Windows)
Detekce DTMF tónů	DTMF číslice 0 až 9, *, #, A, B, C, D
Minimální délka tónu	40 ms, může být softwarově prodloužen
Minimální prodleva mezi stejnými tóny	40 ms
Minimální prodleva mezi stejnými tóny	0 ms
Odstup signál/šum	10 dB (vyjádřené k nižší amplitudě)
Detekce obsazovacího tónu	základní nastavení pro detekci, 74 z 76 unikátních tónů používaných v 97 zemích podle specifikace CCITT Rec E., Suppl #2.
Přesnost pozitivní detekce hlasu	>98% založeno na testech na databázi reálných světových hovorů
Rychlost pozitivní detekce hlasu	menší než 1/10 sekundy
tónová volba	DTMF číslice 0 až 9, *, #, A, B, C, D; 16 čísel pomocí Bellcore LSSGR Sec 6, TR-NWT-000506 MF číslice 0 až 9, KP, ST, ST1, ST2, ST3
Rychlost	10 číslic/s max., nastavitelné
Pulzní volba	10 čísel 0 až 9
Pulzní frekvence	10 pulzů/s, jmenovitá, nastavitelná
Parametrem	